**FSD INTERVIEW QUESTIONS**

1. [**What is Full Stack development?**](https://mindmajix.com/full-stack-developer-interview-questions#full-stack)

**Full Stack development involves developing both the front end and back end of the web application/website at the same time. This process includes three layers:**

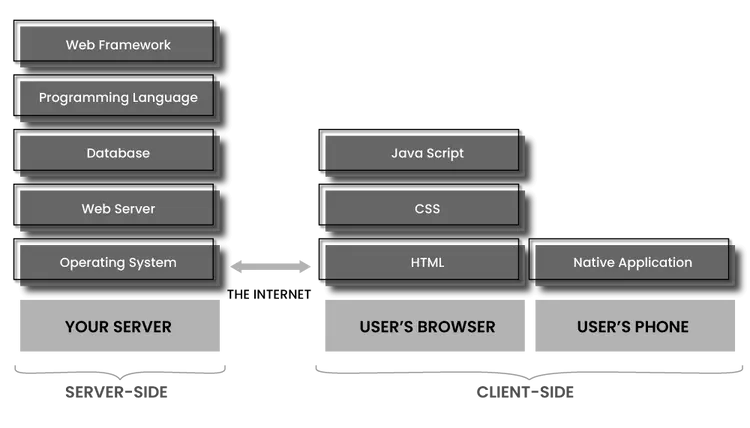
* **Presentation layer (frontend part responsible for user experience)**
* **Business logic layer (backend part refers to the server side of the application)**
* **Database layer.**

1. [**What do Full Stack Web Developers do?**](https://mindmajix.com/full-stack-developer-interview-questions#do)

**A Full Stack Web Developer is a person who is familiar with developing both client and server software. In addition to mastering CSS and HTML, they are also know how to program browsers, databases, and servers.**

**To fully comprehend the role of Full Stack developer, you must understand the web development components - front end and back end**

**The front end comprises a visible part of the application in which the user interacts, while the back end includes business logic.**



1. [**Name a few Full Stack developer tools?**](https://mindmajix.com/full-stack-developer-interview-questions#tools)

**Some of the popular tools used by full-stack developers to make development more accessible and efficient are:**

* **Backbone**
* **Visual Studio Code**
* **WebStorm**
* **Slack**
* **Electron**
* **TypeScript**
* **CodePen**
* **GitHub**

1. [**What skills do you need to be a full-stack developer?**](https://mindmajix.com/full-stack-developer-interview-questions#developer)

**Basic languages - Must be proficient in basic languages like HTML, CSS, and SQL.**

**Front-end frameworks - BootStrap, AngularJS, VueJS,** [**ReactJS**](https://mindmajix.com/react-js-tutorial)**, JavaScript, TypeScript, Python, Ruby, PHP**

**Back-end frameworks - Express, Django, NodeJS, Ruby on Rails**

**Databases - MySQL, SQLite, Postgres,** [**MongoDB**](https://mindmajix.com/what-is-mongodb)**, Cassandra, Apache storm, Sphinx**

**Additional skills recommended - Git, Machine Learning, SSH, Linux Command,** [**Data Structures**](https://mindmajix.com/data-structures-interview-questions)**, Character encoding.**

1. [**Explain Pair Programming?**](https://mindmajix.com/full-stack-developer-interview-questions#pair)

**As the name suggests, Pair Programming is where two programmers share a single workstation. Formally, one programmer at the keyboard called the "driver" writes the code. The other programmer is the "navigator" who views each line of the code written, spell check, and proofread it. Also, programmers will swap their roles every few minutes and vice-versa.**

1. [**What is CORS?**](https://mindmajix.com/full-stack-developer-interview-questions#cors)

**Cross-origin resource sharing (CORS) is a process that utilizes additional HTTP headers to tell browsers to provide a web application running at one origin. CORS accesses various web resources on different domains. Web scripts can be integrated using CORS when it requests a resource that has an external origin (protocol. Domain, or port) from its own.**

1. [**What is Inversion of Control (IoC)?**](https://mindmajix.com/full-stack-developer-interview-questions#inversion-of-control)

**Inversion of Control (IoC) is a broad term used by software developers for defining a pattern that is used for decoupling components and layers in the system. It is mostly used in the context of object-oriented programming. Control of objects or portions of a program is transferred to a framework or container with the help of Inversion of Control. It can be achieved using various mechanisms such as service locator pattern, strategy design pattern, factory pattern, and dependency injection.**

1. [**What is Continuous Integration?**](https://mindmajix.com/full-stack-developer-interview-questions#continuous-integration)

[**Continuous Integration**](https://mindmajix.com/continuous-integration-with-jenkins) **(CI) is a practice where developers integrate code into a shared repository regularly to detect problems early. CI process involves automatic tools that state new code's correctness before integration. Automated builds and tests verify every check-in.**

1. [**How is GraphQL different from REST?**](https://mindmajix.com/full-stack-developer-interview-questions#graphql)

**This is typically a difficult question to answer, but a good developer will be able to go through this with ease. The core difference is GraphQL doesn't deal with dedicated resources. The description of a particular resource is not coupled to the way you retrieve it. Everything referred to as a graph is connected and can be queried to application needs**

1. [**What is the Observer pattern?**](https://mindmajix.com/full-stack-developer-interview-questions#observer-pattern)

**The purpose of the Observer pattern is to define a one-to-many dependency between objects, as when an object changes the state, then all its dependents are notified and updated automatically. The object that watches on the state of another object is called the observer, and the object that is being watched is called the subject.**

1. **What is REST API?**

**REST stands for representational state transfer. A RESTful API (also known as REST API) is an architectural style for an application programming interface (API or web API) that uses HTTP requests to obtain and manage information. That data can be used to POST, GET, DELETE, and OUT data types, which refers to reading, deleting, creating, and operations concerning services.**

1. **What is HTML?**

**HTML stands for HyperText Markup Language and is the language of the internet. It is the standard text formatting language used for creating and displaying pages on the Internet**

**HTML documents are made up of the elements and the tags that format it for proper display on pages.**

1. **What are HTML tags?**

**We use HTML tags for placing the elements in the proper and appropriate format. Tags use the symbols <, and > to set them apart from the HTML content.**

**The** [**HTML tags**](https://www.simplilearn.com/tutorials/html-tutorial/html-tags) **need not be closed always. For example, in the case of images, the closing tags are not required as <img> tag.**

1. **What are HTML Attributes?**

**Attributes are the properties that can be added to an HTML tag. These attributes change the way the tag behaves or is displayed. For example, a <img> tag has an src attribute, which you use to add the source from which the image should be displayed.**

**We add attributes right after the name of the HTML tag, inside the brackets. We can only add the attributes to opening or self-closing tags, but never be in closing tags.**

1. **What is a marquee in HTML?**

**Marquee is used for scrolling text on a web page. It scrolls the image or text up, down, left, or right automatically. To apply for a marquee, you have to use </marquee> tags.**

1. **Define the list types in HTML?**

**The list types in HTML are as below:**

* **Ordered list–The ordered list uses <ol> tag and displays elements in a numbered format.**
* **Unordered list–The unordered list uses <ul> tag and displays elements in a bulleted format.**
* **Definition list–The definition list uses <dl>, <dt>, <dd> tags and displays elements in definition form like in a dictionary.**

1. **Differentiate between an Ordered list and an Unordered list?**

**An unordered list uses <ul> </ul> tags and each element of the list is written between <li> </li> tags. The list items are displayed as bullets rather than numbers.**

**An ordered list uses <ol> </ol> tags and each element of the list is written between <li> </li> tags. The list items are displayed as numbers rather than bullet points.**

**<!DOCTYPE html>**

**<html>**

**<body>**

**<h2>HTML List Example</h2>**

**<ul>**

**<li>Coffee</li>**

**<li>Tea</li>**

**<li>Milk</li>**

**</ul>**

**<ol>**

**<li>Coffee</li>**

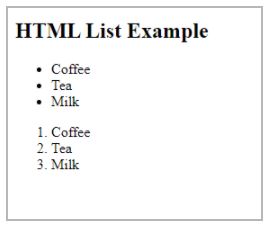
**<li>Tea</li>**

**<li>Milk</li>**

**</ol>**

**</body>**

**</html>**

****

1. **How to create a hyperlink in HTML?**

**We use the anchor tag <a> to create a hyperlink in HTML that links one page to another page.**

**syntax:**

**<a href=”www.novitech.com”>NoviTech</a>**

1. **Which HTML tag is used to display the data in the tabular form?**

**table : In HTML “table” tag is used to define a table.**

**tr : For defining table rows, the “tr” tag is used.**

**th : “th” stands for a table header, for defining table header “th” tag is used. A table header is bold and centred.**

**td : To insert or add data in each cell, Table Data “td” tag is used.**

1. **What is the Box model in CSS? Which CSS properties are a part of it?**

**A rectangle box is wrapped around every HTML element. The box model is used to determine the height and width of the rectangular box. The CSS Box consists of Width and height (or in the absence of that, default values and the content inside), padding, borders, margin.**

**Content: Actual Content of the box where the text or image is placed.**

**Padding: Area surrounding the content (Space between the border and content).**

**Border: Area surrounding the padding.**

**Margin: Area surrounding the border.**

1. **What are the advantages of using CSS?**

**Separation of content from presentation - CSS provides a way to present the same content in multiple presentation formats in mobile or desktop or laptop.**

**Easy to maintain - CSS, built effectively can be used to change the look and feel complete by making small changes. To make a global change, simply change the style, and all elements in all the web pages will be updated automatically.**

**Bandwidth - Used effectively, the style sheets will be stored in the browser cache and they can be used on multiple pages, without having to download again.**

1. **What are the limitations of CSS?**

**Browser Compatibility: Some style selectors are supported and some are not. We have to determine which style is supported or not using the @support selector).**

**Cross Browser issue: Some selectors behave differently in a different browser).**

**There is no parent selector: Currently, Using CSS, you can’t select a parent tag.**

1. **How to include CSS in the webpage?**

**1 - External Style Sheet: An external file linked to your HTML document: Using link tag, we can link the style sheet to the HTML page.**

**<link rel="stylesheet" type="text/css" href="mystyles.css" />**

**2 - Embed CSS with a style tag: A set of CSS styles included within your HTML page.**

**<style type="text/css">**

**/\*Add style rules here\*/**

**</style>**

**Add your CSS rules between the opening and closing style tags and write your CSS exactly the same way as you do in stand-alone stylesheet files.**

**3 - Add inline styles to HTML elements(CSS rules applied directly within an HTML tag.): Style can be added directly to the HTML element using a style tag.**

**<h2 style="color:red;background:black">Inline Style</h2>**

**4 - Import a stylesheet file (An external file imported into another CSS file): Another way to add CSS is by using the @import rule. This is to add a new CSS file within CSS itself.**

**@import "path/to/style.css";**

### Differentiate between CSS3 and CSS2.

**The main difference between** [**CSS3**](https://www.simplilearn.com/css3-resources-ultimate-list-article) **and CSS2 is that CSS divides different sections into modules and supports many browsers. It also contains new General Sibling Combinators responsible for matching similar elements.**

1. **List the CSS Frameworks.**

**CSS frameworks are libraries that make web page styling easier. Some of them are Foundation, Bootstrap, Gumby, Ukit, Semantic UI, etc.**

### What is meant by RGB stream?

**RGB represents colors in CSS. The three streams are namely Red, Green, and Blue. The intensity of** [**colors**](https://www.simplilearn.com/tutorials/css-tutorial/css-colors) **is represented using numbers 0 to 256. This allows CSS to have a spectrum of visible colors.**

### Define z-index.

**This is one of the most frequently asked CSS interview questions. Z-index is used to specify the stack order of elements that overlap each other. Its default value is zero and can take both negative and positive values. A higher z-index value is stacked above the lower index element. It takes the following values- auto, number, initial, and inherit.**

### What is the difference between inline, inline-block, and block?

**Block Elements are <div> and <p>. They usually start on a new line and can take space for an entire row or width.**

**Inline elements are <a>, <span>, <strong>, and <img> tags. They don't start on a new line. However, they appear on the same line as the content and tags beside them.**

**Inline block elements have padding and margins and set height and width values. Though, they are similar to inline elements.**

### What are Pseudo classes?

**Pseudo-classes are the type of pseudo-elements that don’t exist in a normal document tree. It allows selecting the regular elements under certain conditions especially when we try to hover over the link; the anchor tags are :link, :visited, :hover, :active, :focus**

**In this example, the color will be red on the anchor tag when it’s hovered.**

**/\* mouse over link \*/**

**a:hover {**

**color: #FFOOFF;**

**}**

### What is VH/VW (viewport height/ viewport width) in CSS?

**VH and VW are CSS units used to measure viewport height and viewport width respectively in percentage form in the responsive design techniques. E.g. If the height of the browser is 1000px, then VH is 1/100 of the height of the viewport that is 1000px\*(1/100) = 10px, which is the height of the browser. The same applies to VW (viewport width).**

### How is the border-box different from the content box?

**Border-box consists of properties such as content, padding, and the border with respect to height and width. However, Content-box is a default value property used for the box-sizing as well as it helps to add border and padding to give proper height and width to the box without having a border and padding properties**

### What is the grid system?

**The CSS grid system is a type of powerful layout of 2 dimensional systems with respect to columns and rows.**

### How does absolute positioning work?

### Absolute positioning is used to place the element which is then removed from the HTML document from the normal workflow without creating any space for the element in the HTML page layout. The element can be positioned respectively to the closest positioned ancestor; otherwise, if the ancestor is not found, the element is placed with respect to the initial container box. The values provided to the top, right, left and bottom determine the final position of the element.

1. **What does the CSS float property do?**

**The float property of CSS positions an image to the right or left as needed, including text wrapping around it. All properties of elements used before it remain unchanged.**

### What are the properties of flexbox?

**The properties of flexbox are flex-direction, wrap, flow, content, and align-items, and content.**

1. **What is JavaScript?**

**JavaScript is a very powerful client-side scripting language. JavaScript is used mainly for enhancing the interaction of a user with the webpage. In other words, you can make your webpage more lively and interactive, with the help of JavaScript. JavaScript is also being used widely in game development and Mobile application development.**

1. **What are the different data types present in javascript?**

**1. Number**

**2. String**

**3. Boolean**

**4. Undefined**

**5. Null**

**6. Symbol (ES6 and later)**

**7. Object**

**8. BigInt (ES11 and later)**

**9. Function**

1. **Explain Hoisting in javascript.**

**Hoisting is a JavaScript behavior where variable and function declarations are moved to the top of their containing scope during the compilation phase. This allows you to use variables and functions before they are declared in the code.**

1. **Difference between var and let keyword in javascript.**

**var has a function-level scope, while let has a block-level scope.**

**Variables declared with var are hoisted to the top of their scope, while let variables are hoisted but not initialized.**

**let allows you to create block-scoped variables, which helps prevent issues related to variable hoisting and redeclaration.**

1. **What is the difference between == and === operators**

**== is the equality operator that performs type coercion, allowing comparison of values of different types after converting them to a common type.**

**=== is the strict equality operator that checks both value and type, ensuring that values are of the same type without performing type coercion.**

1. **Name the types of functions**

**Function Declaration**

**Function Expression**

**Arrow Function**

**IIFE (Immediately Invoked Function Expression)**

1. **What is the difference between JavaScript and JScript?**

**JavaScript is the scripting language standardized by ECMA International and widely supported in web browsers.**

**JScript is Microsoft's implementation of ECMAScript, used primarily in Internet Explorer.**

1. **How to write a hello world example of JavaScript?**

**console.log('Hello, World!');**

1. **What is scope in javascript**

**Scope refers to the visibility and accessibility of variables in a particular context during runtime.**

**JavaScript has two types of scope: global scope and local scope.**

**Variables declared with var have function-level scope, while those declared with let and const have block-level scope.**

1. **What are the key differences between Java and JavaScript? / How is JavaScript different from Java?**

**Java is a programming language used for server-side development, while JavaScript is a scripting language primarily used for client-side web development.**

**Java requires compilation, while JavaScript is an interpreted language.**

**Java is statically typed, whereas JavaScript is dynamically typed.**

**Java is primarily used for building standalone applications and enterprise-level systems, while JavaScript is widely used for enhancing the interactivity of web pages.**

1. **Is JavaScript case sensitive language?**

**Yes, JavaScript is a case-sensitive language.**

1. **What is DOM? What is the use of document object?**

**The DOM is a programming interface for web documents.**

**It represents the structure of a document as a tree of objects, where each object corresponds to a part of the document.**

**The Document object represents the entire HTML or XML document.**

1. **What is the use of window object?**

**The document object in JavaScript represents the web page or document loaded in the browser.**

**It provides methods and properties to interact with the document, such as modifying its content, structure, and styles.**

1. **How to write HTML code dynamically using JavaScript?**

**// Example: Dynamically create a paragraph element and append it to the body**

**var paragraph = document.createElement('p');**

**var text = document.createTextNode('Hello, dynamically created HTML!');**

**paragraph.appendChild(text);**

**document.body.appendChild(paragraph);**

1. **How to create an array in JavaScript?**

**// Example: Creating an array**

**var fruits = ['apple', 'orange', 'banana'];**

1. **Difference between Client side JavaScript and Server side JavaScript**

**Client-side JavaScript runs in a user's web browser and is used for enhancing the interactivity of web pages.**

**Server-side JavaScript runs on the server and is used for tasks such as handling requests, accessing databases, and performing server-side logic.**

1. **What is the real name of JavaScript?**

**The real name of JavaScript is ECMAScript.**

1. **What is the difference between undefined value and null value?**

**undefined is a variable that has been declared but has not been assigned a value.**

**null is an assignment value representing the intentional absence of any object value.**

1. **What is the requirement of debugging in JavaScript?**

**Debugging in JavaScript is necessary to identify and fix errors or bugs in the code.**

**It helps developers understand the flow of the program, inspect variable values, and catch unexpected behaviors.**

1. **What is ReactJS?**

**ReactJS is a JavaScript library for building user interfaces.**

**It allows developers to create reusable UI components and efficiently update and render components based on changes in data.**

1. **Why ReactJS is Used?**

**ReactJS is used for building user interfaces, especially for single-page applications where dynamic content and seamless updates are crucial.**

**It allows developers to create reusable components, making the code modular and maintainable.**

**React facilitates efficient rendering of components by using a virtual DOM, improving performance.**

1. **What are the features of ReactJS?**

**Virtual DOM: Efficiently updates and renders components for improved performance.**

**Component-Based Architecture: Encourages reusability and maintainability.**

**Unidirectional Data Flow: Ensures predictable data flow within the application.**

**JSX: Provides a syntax extension for JavaScript, making it easier to write UI components.**

**React Hooks: Allows the use of state and other React features in functional components.**

1. **How to create components in ReactJS?**

**Components in ReactJS can be created using either class components or functional components.**

**Example of a functional component:**

**import React from 'react';**

**const MyComponent = () => {**

**return <div>Hello, React!</div>;**

**};**

**export default MyComponent;**

1. **What is JSX?**

**JSX (JavaScript XML) is a syntax extension for JavaScript.**

**It allows the mixing of HTML-like code within JavaScript, making it easier to write React components.**

**Example:**

**const element = <h1>Hello, JSX!</h1>;**

1. **How is React different from React Native?**

**React: Used for building web applications. Components are rendered in the browser.**

**React Native: Used for building mobile applications. Components are rendered using native APIs for iOS and Android platforms.**

**React Native allows the use of React components for building mobile UIs, providing a bridge between JavaScript and native code.**

1. **What is the virtual DOM?**

**The virtual DOM (Document Object Model) is a lightweight, in-memory representation of the actual DOM.**

**React uses the virtual DOM to efficiently update and render components by minimizing direct manipulation of the actual DOM, which can be a resource-intensive operation.**

1. **Why use React instead of other frameworks, like Angular?**

**React's component-based architecture promotes reusability and maintainability.**

**It provides a more flexible and efficient approach to rendering through the use of a virtual DOM.**

**React is often chosen for its simplicity, allowing developers to integrate it into existing projects without significant rewrites.**

1. **What is an event in React?**

**In React, events are interactions triggered by users, such as clicking a button or typing in an input field.**

**Events in React are handled using synthetic event wrappers, providing consistent behavior across different browsers.**

**Example:**

**<button onClick={handleClick}>Click me</button>**

1. **What are forms in React?**

**Forms in React are used to handle user input.**

**React provides a controlled component approach, where form elements are controlled by React state.**

**const [inputValue, setInputValue] = useState('');**

**const handleChange = (e) => {**

**setInputValue(e.target.value);**

**};**

**<input type="text" value={inputValue} onChange={handleChange} />**

1. **How is React different from Angular?**

**React:**

* + **Library for building user interfaces.**
  + **Uses a virtual DOM for efficient rendering.**
  + **Follows a unidirectional data flow.**
  + **Provides JSX for writing components.**

**Angular:**

* + **Full-fledged framework for building web applications.**
  + **Two-way data binding is a core feature.**
  + **Uses a real DOM.**
  + **Requires a steeper learning curve compared to React.**

1. **What is the use of render() in React?**

**The render() method in React is responsible for rendering the component's UI.**

**It returns a description of what the UI should look like, based on the component's current state and props.**

1. **What is a state in React?**

**State in React is an object that represents the internal data of a component.**

**Changes to the state trigger re-rendering of the component, updating the user interface.**

**State is mutable and managed using the setState() method.**

1. **What are props in React?**

**Props (short for properties) are inputs that are passed into a React component.**

**Props allow data to be passed from a parent component to a child component.**

**Props are immutable and should not be modified within the component.**

1. **What are the differences between state and props?**

**State:**

* + **Managed within the component.**
  + **Can be changed using setState().**
  + **Local to the component.**

**Props:**

* + **Received from a parent component.**
  + **Immutable and should not be changed.**
  + **External to the component.**

1. **What are the differences between class and functional components?**

**Class Components:**

* + **Can have state and lifecycle methods.**
  + **Defined using ES6 class syntax.**
  + **Requires the use of this keyword to access props and state.**

**Functional Components:**

* + **Primarily used for presenting UI.**
  + **Introduced in React as a simpler way to write components.**
  + **Cannot have state or lifecycle methods until the introduction of Hooks in React 16.8.**

1. **Explain the difference between frontend and backend development?**

**Frontend Development:**

* + **Focuses on the user interface and user experience of a website or application.**
  + **Involves building and designing elements that users interact with directly.**
  + **Technologies include HTML, CSS, and JavaScript.**

**Backend Development:**

* + **Concerned with server-side operations, databases, and the overall logic of the application.**
  + **Involves server-side scripting, database management, and server configuration.**
  + **Technologies include server-side languages like Node.js, Python, PHP, and databases like MySQL or MongoDB.**

1. **Explain the purpose of the backend?**

**The backend serves to handle and process data, perform business logic, and interact with databases.**

**It manages the server, authentication, authorization, and ensures the overall functionality of an application.**

1. **What is the difference between a GET and a POST request?**

**GET Request:**

* + **Retrieves data from a specified resource.**
  + **Parameters are included in the URL.**
  + **Limited data transfer capacity.**
  + **Cached by browsers.**

**POST Request:**

* + **Submits data to be processed to a specified resource.**
  + **Parameters are sent in the request body.**
  + **Suitable for larger data transfers.**
  + **Not cached by browsers.**

1. **What is a web server?**

**A web server is software that handles HTTP requests from clients, usually web browsers.**

**It processes these requests, retrieves the requested resources, and sends them back to the client.**

**Examples include Apache, Nginx, and Microsoft IIS.**

1. **What is API?**

**An API is a set of rules and tools that allows different software applications to communicate with each other.**

**It defines the methods and data formats that applications can use to request and exchange information.**

**APIs are used to enable the integration of different systems and services.**

1. **What is the difference between API and RESTful API?**

**API (Application Programming Interface):**

* + **A general term referring to a set of rules allowing different software applications to communicate.**
  + **It can follow various architectural styles, including RESTful principles.**

**RESTful API:**

* + **A specific type of API adhering to the principles of Representational State Transfer (REST).**
  + **Utilizes standard HTTP methods (GET, POST, PUT, DELETE) for communication and resource manipulation.**

1. **What is Node.js? Where can you use it?**

**Definition:**

* + **Node.js is a server-side JavaScript runtime environment.**

**Use:**

* + **It is used for building scalable and high-performance network applications.**
  + **Commonly used for developing server-side applications, especially web servers.**

1. **Why use Node.js?**

**Non-blocking I/O:**

* + **Handles concurrent connections efficiently without blocking the execution of code.**

**JavaScript:**

* + **Enables full-stack development using a single programming language.**

**Vibrant Ecosystem:**

* + **Large community and a rich ecosystem of packages (npm) for easy integration of libraries and tools.**

1. **How does Node.js work?**

**Event-Driven and Asynchronous:**

* + **Operates on an event-driven, non-blocking model.**
  + **Utilizes callbacks to handle asynchronous operations.**

**Single-Threaded:**

* + **Uses a single-threaded event loop to handle multiple concurrent connections efficiently.**
  + **Delegates I/O operations to the operating system, freeing the main thread.**

1. **Explain callback in Node.js.**

**Definition:**

* + **A callback in Node.js is a function passed as an argument to another function, to be executed later when a specific operation has been completed.**

**Purpose:**

* + **Used to handle asynchronous operations, ensuring that the code doesn't wait for the operation to complete before moving on.**
  + **Enables non-blocking I/O in Node.js.**

1. **What is NPM?**

**NPM is the default package manager for Node.js.**

**It allows developers to discover, install, and manage packages (libraries and tools) for their Node.js projects.**

1. **What are the modules in Node.js?**

**Modules in Node.js are encapsulated units of functionality that can be reused in different parts of a program.**

**Common types include core modules (built-in), local modules (user-created), and third-party modules (installed via NPM).**

1. **Is Node.js cross-platform?**

**Yes, Node.js is cross-platform.**

**It can run on various operating systems, including Windows, macOS, and Linux, providing a consistent development environment.**

1. **Does Node.js use JavaScript?**

**Yes, Node.js uses JavaScript as its primary programming language.**

**It extends the use of JavaScript from the browser to server-side development.**

1. **Name types of API functions supported by Node.js.**

**Node.js supports various types of API functions, including:**

**File System API: For file-related operations.**

**HTTP API: For creating web servers and handling HTTP requests.**

**Net API: For creating network-related applications.**

**Crypto API: For cryptographic operations.**

**Process API: For interacting with the system's processes.**

1. **What is a database?**

**A database is a structured collection of data organized for efficient storage, retrieval, and management.**

1. **What is a database system?**

**A database system is a software application or suite of applications that facilitates the creation, management, and manipulation of databases.**

1. **What is RDBMS?**

**RDBMS is a type of database management system that organizes data into tables with rows and columns, establishing relationships between them.**

1. **How many types of database languages are?**

**Two main types of database languages are:**

**Data Definition Language (DDL): Defines and manages the structure of the database, including creating, altering, and deleting tables and relationships.**

**Data Manipulation Language (DML): Deals with the manipulation of data stored in the database, including querying, inserting, updating, and deleting records.**

1. **What is MongoDB ?**

**MongoDB is a NoSQL database management system.**

**It stores data in flexible, JSON-like documents, allowing for dynamic and scalable data schemas.**

**MongoDB is known for its ability to handle large amounts of unstructured or semi-structured data and its horizontal scalability.**

1. **What is a Document in MongoDB?**

**In MongoDB, a document is a basic unit of data storage.**

**It is a JSON-like BSON (Binary JSON) object that contains key-value pairs, where values can be various data types, including arrays and other documents.**

**Documents are organized into collections within MongoDB.**

1. **What is the Mongo Shell?**

**The Mongo Shell is a command-line interface for interacting with MongoDB.**

**It allows users to connect to a MongoDB server, execute queries, and perform administrative tasks using JavaScript-like commands.**

1. **What does SQL in MySQL stand for?**

**In MySQL, SQL stands for "Structured Query Language."**

**SQL is a standard language used for managing and manipulating relational databases.**

1. **What are some of the common MySQL commands?**

**SELECT: Retrieve data from one or more tables.**

**INSERT: Add new records into a table.**

**UPDATE: Modify existing records in a table.**

**DELETE: Remove records from a table.**

**CREATE TABLE: Create a new table with specified columns and data types.**

**ALTER TABLE: Modify the structure of an existing table.**

**DROP TABLE: Delete a table and its data.**

**SHOW DATABASES: Display a list of available databases.**

1. **What is the MySQL server?**

**The MySQL server is a relational database management system (RDBMS) that stores and manages structured data.**

**It handles client requests, executes SQL queries, and manages databases and tables.**

1. **What are the data types of MongoDB?**

**MongoDB supports various data types, including:**

* **String**
* **Integer**
* **Double**
* **Boolean**
* **Object**
* **Array**
* **Null**
* **Date**
* **ObjectId (a 12-byte identifier typically employed as a primary key)**

1. **Differentiate MongoDB and MySQL?**

**MongoDB:**

* + **NoSQL database.**
  + **Document-oriented.**
  + **Schema-free.**
  + **Uses BSON (Binary JSON) for data representation.**
  + **Horizontal scalability.**

**MySQL:**

* + **SQL database.**
  + **Relational database management system (RDBMS).**
  + **Schema-based.**
  + **Uses tables with predefined schemas.**
  + **Vertical scalability.**

1. **Difference between SQL and NOSQL?**

**SQL (Structured Query Language):**

* + **Relational databases (RDBMS).**
  + **Table-based data representation.**
  + **Predefined schema.**
  + **ACID properties (Atomicity, Consistency, Isolation, Durability).**

**NoSQL (Not Only SQL):**

* + **Non-relational databases.**
  + **Various data models (document, key-value, wide-column, graph).**
  + **Schema-less or dynamic schema.**
  + **CAP theorem (Consistency, Availability, Partition tolerance).**

1. **What do you mean by Data Model?**

**A data model is a conceptual representation of data and the relationships between different elements within a system.**

**It defines how data is organized and stored, providing a blueprint for designing databases.**

1. **What is server?**

**In a general computing context, a server is a computer or software application that provides services or resources to other computers, known as clients, over a network.**

**In the context of databases, a server often refers to the machine or software responsible for managing and responding to database requests, serving as a central point for data storage and retrieval.**

# 